

教育部考试中心指定教材配套辅导
National Computer Rank Examination

全国计算机 等级考试



上机考试习题集

二级C语言程序设计

考试研究中心 主编

TP3-44
42/2:5

附 参考答案
可调试程序

· 中科多媒体电子出版

全国计算机等级考试试题集

二级 C 语言程序设计

考试研究中心 主编

中科多媒体电子出版社

内 容 提 要

本系列根据教育部考试中心 2002 年公布的《全国计算机等级考试大纲》的要求编写。本册的特点是：知识点全且概括精炼，例题分析透彻，提供了大量与真实考试难度及题型相当的考题供考生训练，使读者顺利通过考试。

未经许可，不得以任何方式复制或抄袭光盘及配书之部分或全部内容。
版权所有，翻版必究。

书 名：全国计算机等级考试试题集——二级 C 语言程序设计
电脑制作：北京暴风雪科技有限公司
出 版：中科多媒体电子出版社
印 刷：北京市富生印刷厂
经 销：各地新华书店
开 本：787×1092 1/16 印张：10
版 次：2003 年 8 月第 1 版



凡购买的图书和光盘有问题者，请向购买处调换。

电话：(010) 62522622、13801352164

致读者

经过几年的全国计算机等级考试辅导实践,我们得到了全国广大考生的极大支持。在这几年的过程中,大量的考生纷纷来信、来电或发 e-mail 给我们,为我们提供了极其丰富的、极具价值的各类信息和建议。2003 年 1 月至 4 月,我们对参加全国计算机等级考试的部分考生进行了历时四个月的联合调查*,本次调查共计 3756 人,按地理区域划分涉及华北(29%)、华中(23%)、华南(17%)、东北(14%)、西南(11%)、西北(6%)。

从本次联合调查的结果分析中,我们得到了很多极具价值的结论,其中以下几类问题比较集中:

- 76% 的被访者认为笔试部分相对上机部分较为容易,13% 认为上机部分相对容易,11% 认为无法比较。可见大部分被访者认为“笔试部分通过考试相对容易,而上机部分较难考过”。
- 63% 的被访者对目前市场上的辅导资料不满意,主要表现在错误多、内容严重重复、价格贵等方面。其中对市场上充斥着许多盗印的教材也有反映。
- 82% 的被访者希望能在复习阶段进行模拟上机训练,以适应上机考试的特殊环境。

为了更好地配合广大考生的考前复习,同时根据联合调查的部分结论,我们新推出《全国计算机等级考试——上机习题集》,特点如下:

- ☆ 上机考试环境演示,真实考试环境演示,每一步均有提示,以便考生提前适应考试环境。
- ☆ 提供了大量与真实考试难度及题型相当的考题供考生训练。每套题都配有参考答案,可使考生针对性的进行练习,提高上机考试通过率。
- ☆ 和教材吻合,突出重点难点。

本光盘(二级 C 语言程序设计)的序列号为: 47543-JASJD-LASVI-KSFCK
进行模拟练习时,请勿脱离光盘使用。

* 本次计算机等级考试联合调查的样本从数量上分析具有一定的普遍代表性,但由于地域、受访者职业分布、问卷设计缺陷及调查途径等因素的限制,调查也存在一定的样本缺陷性,以上结论请慎重使用。

目 录

第一部分 上机题解

1.1 考点综述	1
1.2 考点分析	1

第二部分 模拟试题

上机考试模拟题（一）	10
上机考试模拟题（二）	12
上机考试模拟题（三）	15
上机考试模拟题（四）	18
上机考试模拟题（五）	20
上机考试模拟题（六）	23
上机考试模拟题（七）	26
上机考试模拟题（八）	29
上机考试模拟题（九）	33
上机考试模拟题（十）	35
上机考试模拟题（十一）	38
上机考试模拟题（十二）	41
上机考试模拟题（十三）	44
上机考试模拟题（十四）	47
上机考试模拟题（十五）	50
上机考试模拟题（十六）	52
上机考试模拟题（十七）	54
上机考试模拟题（十八）	57
上机考试模拟题（十九）	60
上机考试模拟题（二十）	63
上机考试模拟题（二十一）	66
上机考试模拟题（二十二）	69
上机考试模拟题（二十三）	72
上机考试模拟题（二十四）	75
上机考试模拟题（二十五）	78
上机考试模拟题（二十六）	80
上机考试模拟题（二十七）	83
上机考试模拟题（二十八）	86
上机考试模拟题（二十九）	89

上机考试模拟题 (三十)	92
上机考试模拟题 (三十一)	95
上机考试模拟题 (三十二)	98
上机考试模拟题 (三十三)	101
上机考试模拟题 (三十四)	104
上机考试模拟题 (三十五)	107
上机考试模拟题 (三十六)	110
上机考试模拟题 (三十七)	112
上机考试模拟题 (三十八)	115
上机考试模拟题 (三十九)	118
上机考试模拟题 (四十)	121
上机考试模拟题 (四十一)	123
上机考试模拟题 (四十二)	126
上机考试模拟题 (四十三)	129
上机考试模拟题 (四十四)	131
上机考试模拟题 (四十五)	134
上机考试模拟题 (四十六)	137
上机考试模拟题 (四十七)	140
上机考试模拟题 (四十八)	143
上机考试模拟题 (四十九)	146
上机考试模拟题 (五十)	148

83	(一十) 题组对左季用士
11	(二十) 题组对左季用士
14	(三十) 题组对左季用士
17	(四十) 题组对左季用士
20	(五十) 题组对左季用士
23	(六十) 题组对左季用士
16	(七十) 题组对左季用士
19	(八十) 题组对左季用士
22	(九十) 题组对左季用士
25	(一十二) 题组对左季用士
28	(一十三) 题组对左季用士
31	(一十四) 题组对左季用士
34	(一十五) 题组对左季用士
37	(一十六) 题组对左季用士
40	(一十七) 题组对左季用士
43	(一十八) 题组对左季用士
46	(一十九) 题组对左季用士
49	(二十) 题组对左季用士
52	(二十一) 题组对左季用士
55	(二十二) 题组对左季用士
58	(二十三) 题组对左季用士
61	(二十四) 题组对左季用士
64	(二十五) 题组对左季用士
67	(二十六) 题组对左季用士
70	(二十七) 题组对左季用士
73	(二十八) 题组对左季用士
76	(二十九) 题组对左季用士
79	(三十) 题组对左季用士
82	(三十一) 题组对左季用士
85	(三十二) 题组对左季用士
88	(三十三) 题组对左季用士
91	(三十四) 题组对左季用士
94	(三十五) 题组对左季用士
97	(三十六) 题组对左季用士
100	(三十七) 题组对左季用士
103	(三十八) 题组对左季用士
106	(三十九) 题组对左季用士
109	(四十) 题组对左季用士
112	(四十一) 题组对左季用士
115	(四十二) 题组对左季用士
118	(四十三) 题组对左季用士
121	(四十四) 题组对左季用士
124	(四十五) 题组对左季用士
127	(四十六) 题组对左季用士
130	(四十七) 题组对左季用士
133	(四十八) 题组对左季用士
136	(四十九) 题组对左季用士
139	(五十) 题组对左季用士

第一部分 上机题解

1.1 考点综述

通过分析历年上机的考试题目,我们发现这些试题主要集中考察以下四类问题:

- 以特定格式输出图形(循环的嵌套使用)
- 编写程序实现有关数组的操作。比如:一位数组的排序,寻找二维数组中的最大最小值问题,以及有关字符数组的操作。
- 素数问题
- 常见错误分析
- 其他问题

1.2 考点分析

本章的例题和练习题的程序的答案都在 TurboC2.0 环境下运行,调试通过。

1.2.1 以特定格式输出图形问题

【例 1】请编写程序输出下列图形。

行号

0

1

2

3

0

1

2

```

  *
 ***
*****
*****
*****
  *
 ***
*****

```

像这样中间大,两头小的问题,我们一般把它分成两部分来对待,即向我们前面分析所示,我们将它分成前面 4 行,和后面 3 行分别统计每行的空格数和字符数,找出空格数以及字符数同行号的关系,添加到我们的程序模板中去即可。上例中前 4 行(即 0-3 行),在第 i 行空格数为 $3-i$,字符数为 $2i+1$ 个,后三行(0-2 行)在第 i 行空格数为 $i+1$,字符数为 $5-2i$ 个。添加到程序中即可。

程序模板:格式输出

```
main()
```

```
{ int i,j,k;/*分别纪录行号,空格,字符数*/
```

```
for(i=0;i<= ;i++)
```

```
{
```

```
for(j=0;j< ;j++) printf(" ");
```

```
for(k=0;k< ;k++) printf("*");
```

```
printf("\n");/*每输完一行后换行*/
```

```

    /*前半部分*/
    for(i=0;i<= 2 ;i++)
    {
        for(j=0;j< 3;j++) printf(" ");
        for(k=0;k< 3;k++) printf("*");
        printf("\n");
    }
    /*后半部分*/

```

对于这个例子我们可以改写成这样:

```

main()
{
    int i,j,k; /*分别纪录行号, 空格, 字符数*/
    for(i=0;i<= 3;i++)
    {
        for(j=0;j< 3-i;j++) printf(" ");
        for(k=0;k<2*i+1;k++) printf("*");
        printf("\n"); /*每输完一行后换行*/
    }
    /*前半部分*/
    for (i=0;i<= 2;i++)
    {
        for(j=0;j< 3-i;j++) printf(" ");
        for(k=0;k< 5-2*i;k++) printf("*");
        printf("\n");
    }
    /*后半部分*/
}

```

数组操作问题

这方面的内容主要包括:

➤ 一维数组排序问题

➤ 数组的最大, 最小值问题

➤ 字符数组问题

1.2.2.1 数组排序模型

一般我们采用冒泡法或者选择法。冒泡法是将相邻的两个数比较, 将小(大)的调到前头。选择法是一次将最小(大)的数放到队列的最前面。我们只要掌握一种即可, 我们在这里介绍冒泡法排序。(考点主要在数据的交换语句处, 包括填空题也经常遇到)

【例 2】对于输入的 10 个整数, 按从小到大顺序进行排序。

```

main()
{
    int a[10];
    int i,j,t;
    for(i=0;i<10;i++)

```

```

        printf("%d\t", a[i]);
        for(j=i+1;j<10;j++)
            if(a[i]>a[j])
            {
                t=a[i];
                a[i]=a[j];
                a[j]=t;
            }
    }
}

```

```

scanf("%d",&a[i]);
printf("\n");
for(j=0;j<=8;j++)
{
    for(i=0;i<=9-j;i++)
        if(a[i]>a[i+1])
            {t=a[i];a[i]=a[i+1];a[i+1]=t;}
    /*如果 a[i]>a[i+1],那么将两者进行交换。其中 t 相当于一个临时储存器,进行交换用的。*/
}
for(i=0;i<10;i++)
    printf("%3d",a[i]);
}

```

注意：在考试的时候请务必注意排序的顺序是从大到小，还是从小到大。

1.2.2.2 数组的最大、最小值问题

主要包括：一维、二维数组中最大、最小值的寻找问题（包括纪录所在行号、列号问题）。另外有关寻找鞍点的操作要求大家比较熟悉。下面，我们介绍一下，如何来找出数组中的最大、最小值问题。

我们知道，在寻找一维数组的最大值时，我们可以将数组的第一个元素同时赋给 max、min 变量，然后数组里的每一个元素同 max、min 变量相比较，如果大于 max，就将该变量值赋给 max，如果小于 min，就将该变量值赋给 min。对于二维数组我们也可以这样处理。我们来看下面的例题：

【例 3】已知矩阵 $a[4][4]=\{1,3,5,7,14,11,4,50,12,3,9,15,60,12,8,6,95\}$ ，求最大值。（注意数组名作为函数参数）

```

int qmax(int a[4][4])
{
    int max,i,j;
    max=a[0][0];
    for(i=0;i<4;i++)
        for(j=0;j<4;j++)
            if(a[i][j]>max) max=a[i][j];
    return(max);
}

main()
{
    static int a[4][4]={1,3,5,7,14,11,4,50,12,3,9,15,60,12,8,6,95};
    printf("%d",qmax(a));
}

```

1.2.2.3 字符数组问题

【例 4】字符数组的输入、输出和字符数组中某几个元素个数的统计

程序模板：

```

#include "string.h"

main()
{

```

```

char s[25];
int len,i,m=0,n=0;/*赋初值*/
printf("please input the string\n");
gets(s);
len=strlen(s);
for(i=0;i<len;i++)
{
    if(s[i]=='a') m++;
    if(s[i]=='b') n++;
}
printf("%3d,%3d\n",m,n);
}

```

素数问题

【例 5】请编一个函数 `int fun(int pm)`，它的功能是：判断 `pm` 是否是素数。若 `pm` 是素数，返回 1；若不是素数，返回 0。`pm` 的值由主函数从键盘读入。

请勿改动主函数 `main` 和其它函数中的任何内容，仅在函数 `fun` 的花括号中填入你编写的若干语句。

```

#include <conio.h>
#include <stdio.h>
#include <math.h>
int fun(int a)
{

```

/*答案开始 */

```

:
:

```

/*答案结束 */

```

}

```

`main()` / * 主函数 * /

```

{

```

```

    int x;

```

```

    clrscr();

```

```

    printf("\nPlease enter a number:");

```

```

    scanf("%d", &x);

```

```

    printf("%d\n", fun(x));

```

```

}

```

分析：我们可以添加程序段如下：

```

int fun(int a)
{

```

```

{

```

```

    int i;

```

```

    if(a==2) return 1;

```

```

i=2;
while ((a%i)!=0 && i<=sqrt((float) a)) i++;
if ((a%i)==0)
{
    printf("%d not is a prime!",a); return 0;
}
printf("%d is a prime!",a); return 1;
}

main()
{
    int x;
    clrscr();
    printf("\nPlease enter a number:");
    scanf("%d", &x);
    printf("%d\n", fun(x));
}

```

常见错误分析

1.2.4.1 第一类错误分析

1. 在使用变量前未定义该变量。

例如：

```

main()
{
    int a, b;
    a=1;
    b=2;
    printf(" %d\n", a+b);
}

```

2. 语句后面漏写分号或不该加分号的地方加了分号。

C 语言规定，语句必须以分号结束，分号是 C 语句不可缺少的一部分，这也是和其它高级语言不同的一点。初学者往往容易忽略这个分号。如：

```

x=1;
y=2;

```

又如，在复合语句中漏写最后一个语句的分号：

```
{t=x; x=y; y=t}
```

3. 不该有空格的地方加了空格

例如，在用 `/*...*/` 对 C 程序中的任何部分作注释时，`/` 与 `*` 之间都不应当有空格。

又如，在关系运算符 `<=`, `>=`, `==` 和 `!=` 中，两个符号之间也不允许有空格。

4. 定义或引用数组的方式不对

C 语言规定，在对数组进行定义或对数组元素进行引用时必须要用方括号(对二维数组或多维数组的每一维数据都必须分别用方括号括起来)，例如以下写法都将造成编译时出错：

```
int a(10); int b[5,4];
printf(" %d\n", b[1+2,2]);
```

5. 混淆字符和字符串

C 语言中的字符常量是由一对单引号括起来的单个字符；而字符串常量是用一对双引号括起来的字符序列。字符常量存放在字符型变量中，而字符串常量只能存放在字符型数组中。例如，假设已说明 num 是字符型变量，则以下赋值语句是非法的：

```
num=" 1" ;
```

6. 在引用数组元素或指针变量之前没对其赋初值

例如：

```
main()
{
    int a[6],b;
    b=a[5];
    :
}

main()
{
    int *ptr;
    *ptr=i;
    :
}
```

以上两个程序段在编译时均会出现警告信息。

7. 混淆数组名与指针变量

在 C 语言中，数组名代表数组的首地址，它的值是一个常量，不能被修改。例如，在以下程序段中，用 a++ 是不合法的。

```
main()
{
    int i, a[10];
    for (i=0; i<10; i++)
        scanf(" %d", a++);
    :
}
```

8. 混淆不同类型的指针

若有以下语句：

```
int *p1, a=1;
float *p2;
p1=&a;
```

则赋值语句 p2=p1 是非法的。

9. 混淆指针说明语句中的*号和执行语句中的*号

设有以下说明语句：

```
int *p1, i=1;
则 *p1=&i; 是不合法的。
```

10. 误将函数形参和函数中的局部变量一起定义

例如：

```
fun(x,y)
float x, y, z;
```

```

{
    x++; y++; z=x+y;
    :
}

```

11. 所调用的函数在调用前未定义

```

main()
{
    float a=1 0, b=2 0, c;
    c=fun(a,b);
    :
}

float fun(x, y)
float x, y;
{ x++; y++;
  :
}

```

12. 混淆结构体类型名和结构体变量名

若定义了以下结构体类型 student:

```

struct student
{
    long/int num;
    char name[20];
    int age;
    float score;
};

```

则赋值语句: (student)num=199401; 是错误的

1.2.4.2 第二类错误分析

1. 在用 scanf 函数给普通变量输入数据时, 在变量名前漏写地址运算符&

如: scanf(" %d%d" gxay);

2. 在 scanf 函数调用语句中, 企图规定输入实型数据的小数位

如执行以下语句:

```
scanf(" %6.2f", &a);
```

3. 输入数据时的数据形式与要求不符

用 scanf 函数输入数据时, 必须注意要与 scanf 语句中的对应形式匹配。如:

```
scanf(" %d.%d", &x, &y);
```

若按以下形式输入数据:

24 2,4

是不合法。数据 2 和 4 之间应当有逗号。

4. 输入、输出时的数据类型与所用格式说明符不匹配

例如有以下说明语句:

```
int x=1; float y=2.5;
```

则运行时执行语句

```
printf(" x=%f, y=%d\n", x, y);
```

将给出与原意不符的结果：(在 TURBO C 2.0 下运行，结果为 0.000000,0)

5. 混淆 "=" 和 "=="

在 C 语言中，"=" 赋值运算符，"==" 是关系运算符。

6. 在不该出现分号的地方加了分号

例如：if(x>y);

```
printf(" x is larger than y. \n");
```

7. 对于复合语句，忘记加花括号

例如： i=1; a=0;

```
while (i<=10)
```

```
{ a+=i; i++; }
```

```
printf(" a=%d\n", a);
```

8. 误把数组说明时所定义的元素个数作为最大下标值使用

C 语言规定，引用数组元素时下标从 0 开始，即下标值的下限为 0，而下标的上限值是数组定义时元素个数减 1。

9. 在 switch 语句的各分支中未使用 break 语句

例如：

```
switch(grade)
```

```
{
```

```
case 'A' : printf(" 85 100\n");
```

```
case 'B' : printf(" 70 84\n");
```

```
case 'C' : printf(" 60 69\n");
```

```
case 'D' : printf(" <60\n");
```

```
default:
```

```
printf(" Error\n");
```

```
}
```

10. 混淆 break 语句和 continue 语句的作用

例如：

```
do
```

```
{
```

```
scanf(" %d", &x);
```

```
if(x>0) break;
```

```
printf(" %d\n", x);
```

```
}while(x!=0);
```

11. 使用 ++ 或 -- 运算符时易犯的错误

如：

```
main()
```

```
{
```

```
int a[5]={1, 2, 3, 4, 5}, *p;
p=a;
printf(" %d\n", *(p++));
:
}
```

输出的结果为 1，选择题中经常遇到。

12. 误解形参值的变化会影响实参的值

例如：

```
main()
{
    int a=1, b=3;
    swap(a, b);
    printf(" a=%d, b=%d\n", a, b);
}
swap(int x, int y)
{
    int m;
    m=x; x=y; y=m;
}
```

原意想通过调用 swap 函数使 a 与 b 的值对换，然而，从输出结果来看，并没有达到目的，主要是由于 a, b 作为实参，它不会改变。

其他问题

口令检查函数

【例 6】用户有三次输入口令的机会。三次机会都失败或输入正确口令时，循环结束。正确口令是“abc123”。

```
main()
{
    char str[20];
    int x;
    for(x=0; x<3 && strcmp(str, "abc123"); ++x)
    {
        printf("enter password please:");
        gets(str);
    }
    if (x==3)
    {
        printf("Wrong password"); exit(0);
    }
    printf("Welcome!"); /* log user in ...*/
}
```

第二部分 模拟试题

上机考试模拟题 (一)

Copy 一、DOS 操作题(共 30 分)

- (1) 将考生目录下的 HOLMMONTY 子目录中的文件 STIMPY.FOR 和考生目录下的 FOLK\ORG 子目录中的文件 GAMES.PAS 顺序合并拷贝到考生目录下的 HINDUTMB 子目录中, 文件名为 JEWISH.BAS。
- (2) 在考生目录下的 ADLNHUTV 子目录中建立一个新的子目录 MORTAL。
- (3) 将考生目录下的 BARRYHERB 子目录中的文件 GIANT.DOC 更名为 EHU-MS.TXT。
- (4) 将考生目录下的 SATLE\ROLX 子目录删除。
- (5) 将考生目录下的 NATURE\CIA 子目录中的文件 TOBACCO.FOX 删除。
- (6) 将考生目录下的 OCEAN\WAVES 子目录中的文件 BEAVIS.FMT 设置成具有只读属性的文件。

二、程序修改题 (1 题, 共 30 分)

给定程序 MOD11.C 中函数 fun 的功能是: 先从键盘上输入一个 3 行 3 列矩阵的各个元素的值, 然后输出主对角线元素之和。

请改正函数 fun 中的错误或在横线处填上适当的内容并把横线删除, 使它得出正确的结果。

注意: 不要改动 main 函数, 不得增行或删行, 也不得更改程序的结构!

```
#include <stdio.h>
```

```
int fun()
```

```
{
```

```
int a[3][3], sum;
```

```
int i, j;
```

```
/******found******/
```

```
sum = 0;  
for (i=0; i<3; i++)
```

```
{
```

```
for (j=0; j<3; j++)
```

```
/******found******/
```

```
scanf("%d", &a[i][j]);
```

```
}
```

```
for (i=0; i<3; i++)
```

```
{
```

```
printf("%d", a[i][i]);
```

```
sum = sum + a[i][i];
```

```
}
```

1 2 3
6 5 1
7 9 8

sum

sum = sum + a

scanf

```
printf("Sum=%d\n",sum);
}
```

```
main()
{
    fun();
}
```

三、程序设计题 (1 小题, 计 40 分)

函数 fun 的功能是: 将两个两位数的正整数 a、b 合并形成一个整数放在 c 中。合并的方式是: 将 a 数的十位和个位数依次放在 c 数的百位和个位上, b 数的十位和个位数依次放在 c 数的十位和千位上。

例如, 当 a=45, b=12。调用该函数后, c=2415。

注意: 部分源程序存在文件 PROG1.C 中。数据文件 IN1.DAT 中的数据不得修改。

请勿改动主函数 main 和其它函数中的任何内容, 仅在函数 fun 的花括号中填入你编写的若干语句。

```
#include <conio.h>
#include <stdio.h>
void fun(int a, int b, long *c)
```

```
{ int m, n, p, q;
```

```
    m = b % 10;
```

```
    n = a / 10;
```

```
    p = b / 10;
```

```
    q = a % 10;
```

```
main()
```

```
{ int a, b; long c;
```

```
clrscr();
```

```
printf("Input a, b:");
```

```
scanf("%d%d", &a, &b);
```

```
fun(a, b, &c);
```

```
printf("The result is: %ld\n", c);
```

```
NONO();
```

```
}
```

```
NONO()
```

```
/* 本函数用于打开文件, 输入数据, 调用函数, 输出数据, 关闭文件。 */
```

```
FILE *rf, *wf;
```

```
int i, a, b; long c;
```

```
rf = fopen("in1.dat", "r");
```

```
wf = fopen("bc04.dat", "w");
```

```
for(i = 0; i < 10; i++) {
```

```
    fscanf(rf, "%d%d", &a, &b);
```

$$*C = m * 1000 + n * 100 + p * 10 + q$$

return (*c)

```
int m, n, p, q;
```

```
a = a * 10 + n;
```

```
m = a / 10;
```

```
n = a % 10;
```

```
p = b / 10;
```

```
q = b % 10;
```

$$C = q * 1000 + m * 100 + p * 10 + n * 1;$$

return (*c);